

# ÉQUATIONS DIFFÉRENTIELLES ORDINAIRES DU PREMIER ORDRE

Bernard Dupont

[Bernard.Dupont@univ-lille1.fr](mailto:Bernard.Dupont@univ-lille1.fr)

Dans le domaine difficile des équations différentielles, le traitement des équations différentielles d'ordre 1 (en abrégé : EDO d'ordre 1) se fait au moyen de deux grandes méthodes. La première est la méthode exacte de résolution, qui est ordinairement celle que préfèrent les économistes parce qu'elle fournit une solution explicite au problème. La seconde est une méthode de résolution approchée, basée sur des approximations numériques, moins satisfaisante dans l'absolu mais très suggestive quand on la couple avec une représentation graphique. Dans les deux cas, on dispose de la même commande **dsolve**. Quand on se trouve en présence d'une équation différentielle particulièrement difficile à résoudre voire insoluble, on tentera une approche qualitative du problème, c'est à dire une approche graphique, qui puise dans les ressources du paquetage **DEtools**. Toutes ces méthodes sont programmables mais il existe un outil interactif puissant, appelé ODE Analyzer Assistant, qui permet d'écourter le travail de frappe des commandes.

## ▼ Méthode exacte de résolution

De manière très générale, une EDO d'ordre 1 s'écrit  $F(x, y, y') = 0$ , où  $y$  est une fonction de la variable indépendante  $x$  et  $y'$  est sa dérivée première. Le plus souvent, on traite une équation sous la forme dite résolue  $y' = F(x, y)$ . Le problème de Cauchy consiste à adjoindre à l'équation différentielle une condition initiale  $y(0) = y_0$ .

La commande universelle **dsolve** exige au moins les deux ingrédients suivants : en premier lieu, écriture d'une équation différentielle avec son éventuelle condition initiale et en second lieu désignation de la fonction inconnue.

L'équation différentielle et la fonction inconnue doivent être écrites en respectant la syntaxe Maple : **y(x)** pour la fonction inconnue et **diff(y(x), x)** pour sa dérivée première.

## ▼ Problèmes sans condition initiale

Un problème sans condition initiale demande de chercher l'ensemble des fonctions satisfaisant une EDO donnée. La théorie mathématique est très complète sur les EDO linéaires. En revanche, les EDO non linéaires sont un domaine où rien n'est assuré hormis pour des cas d'école qui sont loin de recouvrir les besoins pratiques.

### ▼ Principes généraux

On utilise la commande **dsolve(eqd, y(x))** où l'équation différentielle **eqd** a pour inconnue la fonction **y(x)**. Pour résoudre  $y(x) + y'(x) = \cos(x)$ , on écrit :

> **restart:**

```
dsolve(y(x)+diff(y(x),x)=cos(x),y(x));
```

$$y(x) = \frac{1}{2} \cos(x) + \frac{1}{2} \sin(x) + e^{-x} \_CI$$

Mais, à bien des égards, il est préférable de préparer l'écriture du premier argument en

assignant l'équation différentielle, voire la fonction inconnue elle-même et sa dérivée. Rappelons au passage qu'il n'est pas possible d'utiliser le raccourci usuel **f'** pour la dérivée car l'apostrophe est réservée dans le code Maple pour désassigner.

```
> f:=y(x);#assignation de la fonction
    fp:=diff(f,x);#assignation de sa dérivée
    eqd:=f+fp=cos(x);#écriture de l'EDO
    dsolve(eqd,f);
```

$$f := y(x)$$

$$fp := \frac{d}{dx} y(x)$$

$$eqd := y(x) + \frac{d}{dx} y(x) = \cos(x)$$

$$y(x) = \frac{1}{2} \cos(x) + \frac{1}{2} \sin(x) + e^{-x} \_CI$$

On remarque que la constante d'intégration est notée ici **\_CI**. Maple utilise toujours une assignation du type "tiret bas+C +entier" pour les constantes d'intégration, d'où le conseil utile : il faut éviter d'attribuer à une expression un nom sous cette forme dans une feuille de travail, faute de quoi des conflits risquent de se produire.

Autre point important. Maple a renvoyé la réponse suivante, proche de l'écriture manuscrite :

$$y(x) = \frac{1}{2} \cos(x) + \frac{1}{2} \sin(x) + e^{-x} \_CI$$

mais, à y regarder de près, il s'agit d'une égalité ce qui fait que la solution n'est ni une expression ni une fonction. En conséquence, il serait naïf d'utiliser tel quel cet output pour poursuivre l'étude de l'équation différentielle. Il est astucieux de récupérer l'expression présentée dans le membre de droite et la fonction correspondante comme suit :

1°) donner systématiquement un nom à la solution, par exemple :

```
> soluce:=dsolve(eqd,f);
    soluce := y(x) = \frac{1}{2} \cos(x) + \frac{1}{2} \sin(x) + e^{-x} \_CI
```

2°) puis donner un nom au seul second membre de l'égalité, par exemple à l'aide de **subs** :

```
> sol_equadif:=subs(soluce,y(x));
    sol_equadif := \frac{1}{2} \cos(x) + \frac{1}{2} \sin(x) + e^{-x} \_CI
```

3°) Si nécessaire, on peut nommer la fonction solution par **unapply** :

```
> f_soluce:=unapply(sol_equadif,x);
    f_soluce := x \to \frac{1}{2} \cos(x) + \frac{1}{2} \sin(x) + e^{-x} \_CI
```

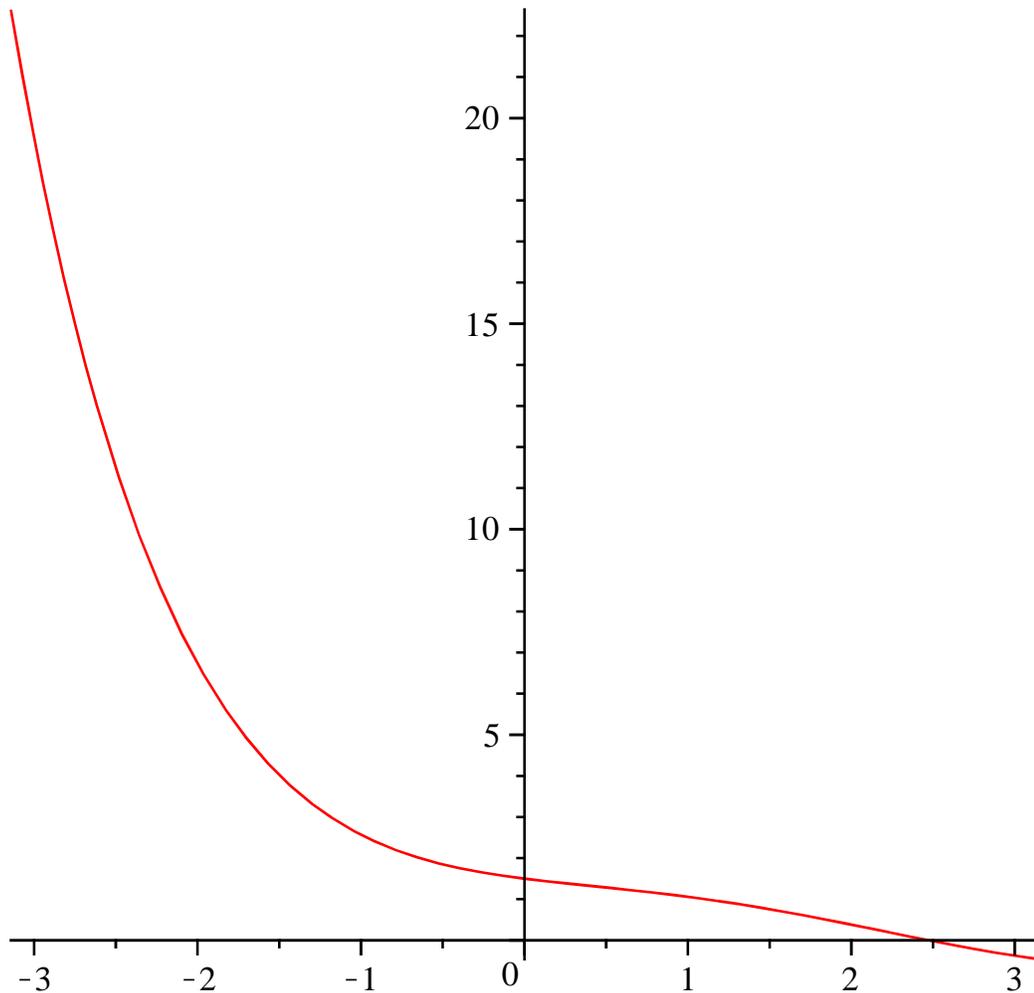
Tout ce qu'on a vu sur les fonctions devient alors possible, à commencer par les calculs de dérivée et la représentation graphique (à condition de donner une valeur à la constante d'intégration) :

```
> diff(sol_equadif,x$3);#calcul de la dérivée d'ordre 3 de
    la fonction solution
    _C1:=1;#attribution d'une valeur à la constante
```

d'intégration pour préparer la représentation graphique  
`plot(f_soluce,-Pi..Pi);`

$$\frac{1}{2} \sin(x) - \frac{1}{2} \cos(x) - e^{-x} \_CI$$

`\_CI := 1`



La commande `dsolve` sait résoudre pratiquement toutes les équations différentielles linéaires (à condition que l'intégration soit possible) et pas mal d'équations bien connues (à variables séparables, Bernouilli, Riccati entre autres).

### ▼ *Equations différentielles linéaires du premier ordre à coefficients constants*

Une EDO linéaire du premier ordre à coefficient constant avec second membre a pour forme générale  $y'(t) + \alpha y(t) = f(t)$  où  $\alpha$  est un réel. La théorie assure que toute équation de ce type est intégrable, ce qu'on peut vérifier aisément :

> `restart; dsolve(diff(y(t),t)+alpha*y(t)=f(t),y(t));`

$$y(t) = \left( \int f(t) e^{\alpha t} dt + \_CI \right) e^{-\alpha t}$$

Evidemment, la solution retournée par Maple dépend de sa réussite en matière d'intégration, donc de la forme du second membre. Dans les modèles économiques, le second membre est généralement constant, ce qui ne pose guère de problème.

```
> eqd:=diff(P(t),t)=-0.05*P(t)+40;
dsolve(eqd,P(t));
```

$$eqd := \frac{d}{dt} P(t) = -0.05 P(t) + 40$$

$$P(t) = 800 + e^{-\frac{1}{20}t} \_CI \quad (1.1.2.1)$$

### Equations différentielles linéaires à coefficients non constants

Une EDO linéaire à coefficient constant a pour forme générale  $y' + \alpha(t) y = f(t)$ . Une fois encore, la théorie assure que toute équation de ce type est intégrable.

```
> restart: eqd:=diff(y(t),t)+alpha(t)*y(t)=f(t); dsolve(eqd,y(t));
```

$$eqd := \frac{d}{dt} y(t) + \alpha(t) y(t) = f(t)$$

$$y(t) = \left( \int f(t) e^{\int \alpha(t) dt} dt + \_CI \right) e^{\int (-\alpha(t)) dt}$$

Mais les problèmes d'intégration sont évidemment plus difficiles en règle générale que dans le cas des EDO linéaires à coefficients constants. Dans les situations "ordinaires", Maple résout sans problème les quadratures. Profitons-en pour rechercher la classe des fonctions à élasticité constante, qui vérifient par définition  $x \frac{f'(x)}{f(x)} = \alpha$  avec  $\alpha$  réel :

```
> dsolve(x*diff(f(x),x)/f(x)=alpha,f(x));
```

$$f(x) = \_CI x^\alpha$$

A une translation près, il s'agit des fonctions "puissance réelle". Moralité : en tout point d'une Cobb-Douglas, un accroissement de 1% de la variable explicative conduit à une variation de  $\alpha$  % de la variable expliquée.

### Equations différentielles non linéaires

On continue notre incursion dans les EDO en intégrant des équations non linéaires. Maple est performant dans un sens précis : si sa bibliothèque reconnaît un cas qui est maîtrisé par la théorie mathématique, il renvoie la solution, sous une forme plus ou moins compréhensible; sinon, il renvoie un écho vide.

Dans ces deux premiers exemples, Maple se joue des difficultés en un temps record.

```
> eqd1:=y(t)*diff(y(t),t)=diff(y(t),t)^2+exp(t); solegg1:=
dsolve(eqd1,y(t));
```

$$eqd1 := y(t) \left( \frac{d}{dt} y(t) \right) = \left( \frac{d}{dt} y(t) \right)^2 + e^t$$

$$solegg1 := y(t) = \frac{1}{\_CI} + \_CI e^t, y(t) = \_CI e^{\frac{1}{2}t}$$

```
> eqd2:=y(t)^2-3*t*y(t)*diff(y(t),t)+t^2*diff(y(t),t)^2=0;
solegg2:=dsolve(eqd2,y(t));
```

$$eqd2 := y(t)^2 - 3ty(t) \left( \frac{d}{dt} y(t) \right) + t^2 \left( \frac{d}{dt} y(t) \right)^2 = 0$$

$$\text{soleqd2} := y(t) = \_CI t^{3/2} t^{-\frac{1}{2}\sqrt{5}}, y(t) = \_CI t^{\frac{1}{2}\sqrt{5}} t^{3/2}$$

Dans ce troisième exemple, la réponse fait référence à la fonction dite W de Lambert, systématiquement invoquée par Maple lorsqu'il doit résoudre une équation algébrique transcendante. En règle générale, dès que les équations différentielles proposées sont hors norme, les outputs peuvent sembler abscons et il faut consulter l'aide en ligne pour comprendre le résultat. Il se peut même qu'on se sente obligé d'approfondir ses connaissances en mathématiques ...

```
> eqd3:=(t^2+t*y(t))*diff(y(t),t)=t^2+y(t)^2;soleqd3:=
dsolve(eqd3,y(t));
```

$$\text{eqd3} := (t^2 + t y(t)) \left( \frac{d}{dt} y(t) \right) = t^2 + y(t)^2$$

$$\text{soleqd3} := y(t) = 2 t \text{LambertW} \left( \frac{1}{2} \frac{e^{-\frac{1}{2}} \_CI}{\sqrt{t}} \right) + t$$

De même, la réponse est décevante dans ce quatrième exemple, qui est en fait l'équation d'accumulation du capital par tête dans un modèle de croissance endogène.

```
> eqd4:=diff(z(t),t)=(p/(b-a))*z(t)-(g/((1-a)*(b-a)))*z(t)^(2-b)+d*(1-a)/(b-a)*z(t)^(1-a);
dsolve(eqd4,z(t));
```

$$\text{eqd4} := \frac{d}{dt} z(t) = \frac{p z(t)}{b-a} - \frac{g z(t)^{2-b}}{(1-a)(b-a)} + \frac{d(1-a) z(t)^{1-a}}{b-a}$$

$$t + \int^{z(t)} \frac{(-1+a)(-b+a)}{-p\_a + p\_a a + g\_a^{2-b} - d\_a^{1-a} + 2 d\_a^{1-a} a - d\_a^{1-a} a^2} d\_a \quad (1.1.4.1)$$

$$+ \_CI = 0$$

Maple donne bien une relation mais c'est au lecteur d'en inférer des informations sur la fonction inconnue. On retrouve le même désagrément dans l'étude du modèle de croissance de Solow où la fonction de production par tête est une Cobb-Douglas :

```
> sol1:=dsolve(diff(k(t),t)=0.2*k(t)^(1/3)-0.05*k(t),k(t));
```

$$\text{sol1} := k(t)^{2/3} - 4 - e^{-\frac{1}{30}t} \_CI = 0$$

L'écho est une solution implicite de l'équation de Solow. Heureusement, l'utilisation de **solve** permet de dégager les solutions explicites :

```
> solve(sol1,k(t));
```

$$\left( 4 + e^{-\frac{1}{30}t} \_CI \right)^{3/2} \quad (1.1.4.2)$$

## ▼ Problèmes avec condition initiale

Il est assez naturel en économie de résoudre une équation différentielle avec une condition initiale dans la mesure où la variable endogène étudiée démarre à une certaine date à un certain niveau connu.

### ▼ Principes généraux

On utilisera la commande `dsolve({equa diff,y(0)=y0},y(t))`. Le premier argument est un ensemble ou une liste formé de l'équation différentielle et de la condition initiale écrite comme une égalité.

```
> dsolve({diff(y(t),t)+2*y(t)=cos(t),y(0)=1},y(t));
```

$$y(t) = \frac{2}{5} \cos(t) + \frac{1}{5} \sin(t) + \frac{3}{5} e^{-2t} \quad (1.2.1.1)$$

En pratique, il convient d'assigner tous les ingrédients. En récupérant la solution, il est possible de représenter graphiquement la trajectoire-solution..

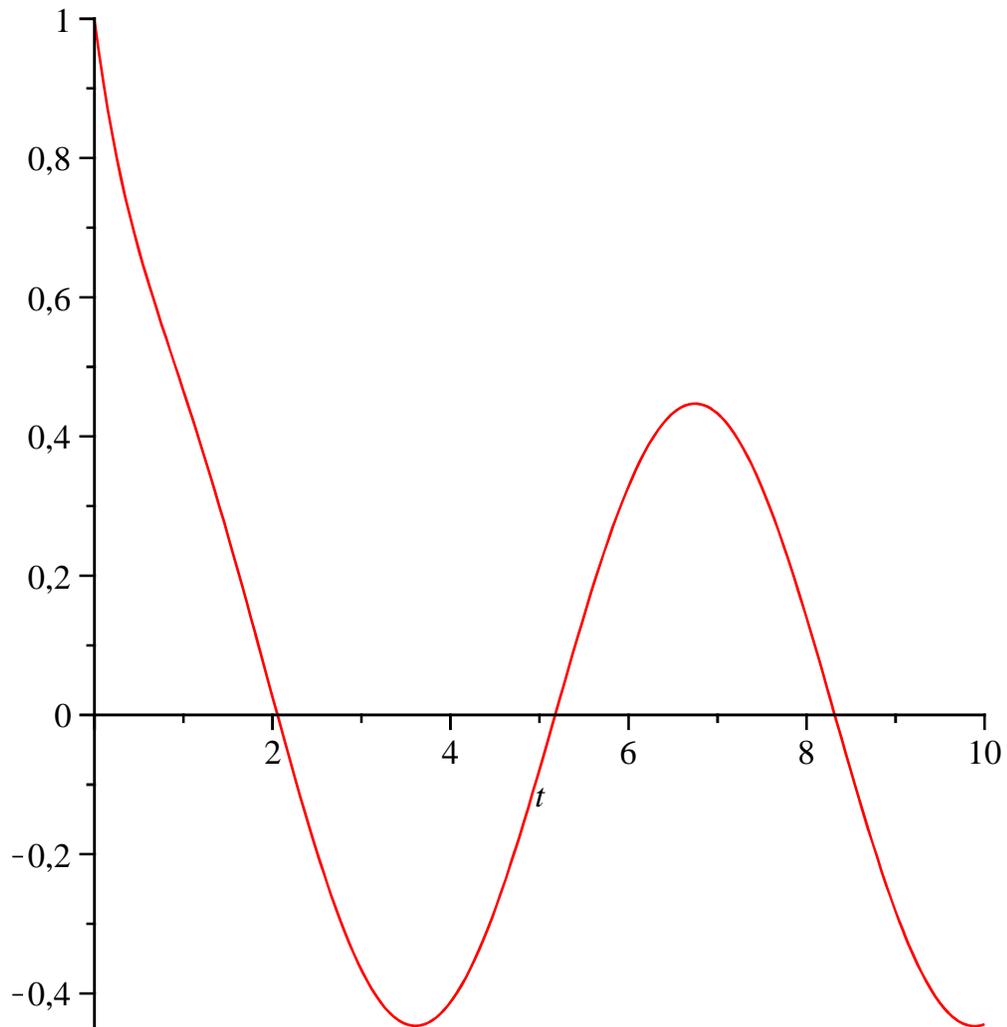
```
> eqd:=diff(y(t),t)+2*y(t)=cos(t);#assignation de l'EDO
cond_init:=y(0)=1;#assignation de la condition initiale
soleqd:=dsolve({eqd,cond_init},y(t));#résolution du
problème de Cauchy
sol:=subs(soleqd,y(t));#récupération de la solution
plot(sol,t=0..10,numpoints=200);#représentation graphique
de la fonction solution
```

$$eqd := \frac{d}{dt} y(t) + 2 y(t) = \cos(t)$$

$$cond\_init := y(0) = 1$$

$$soleqd := y(t) = \frac{2}{5} \cos(t) + \frac{1}{5} \sin(t) + \frac{3}{5} e^{-2t}$$

$$sol := \frac{2}{5} \cos(t) + \frac{1}{5} \sin(t) + \frac{3}{5} e^{-2t}$$



### ▼ *Champ d'application*

En pratique, les problèmes de Cauchy sont tous résolus dans le cas des EDO linéaires à coefficients constants ou variables si les intégrations rencontrées lors du processus de résolution ne sont pas insurmontables (voir exercices M3 et M4). Les équations non linéaires sont évidemment capricieuses puisque la théorie générale de résolution n'existe pas. Il faut malgré tout souligner combien Maple reste performant dans ce domaine. Parfois, la recherche d'une solution peut prendre plusieurs minutes, ou heures, voire ... Il arrive aussi que Maple "sèche" sur certains problèmes (on en donnera un exemple plus bas).

Dans cet exemple, Maple est époustouflant :

```
> restart;
eqd1:=y(t)*diff(y(t),t)=diff(y(t),t)^2+exp(t);#équation
peu sympathique...
soleq1:=dsolve({eqd1,y(0)=1},y(t));#résolution avec une
condition initiale
```

$$eqd1 := y(t) \left( \frac{d}{dt} y(t) \right) = \left( \frac{d}{dt} y(t) \right)^2 + e^t$$

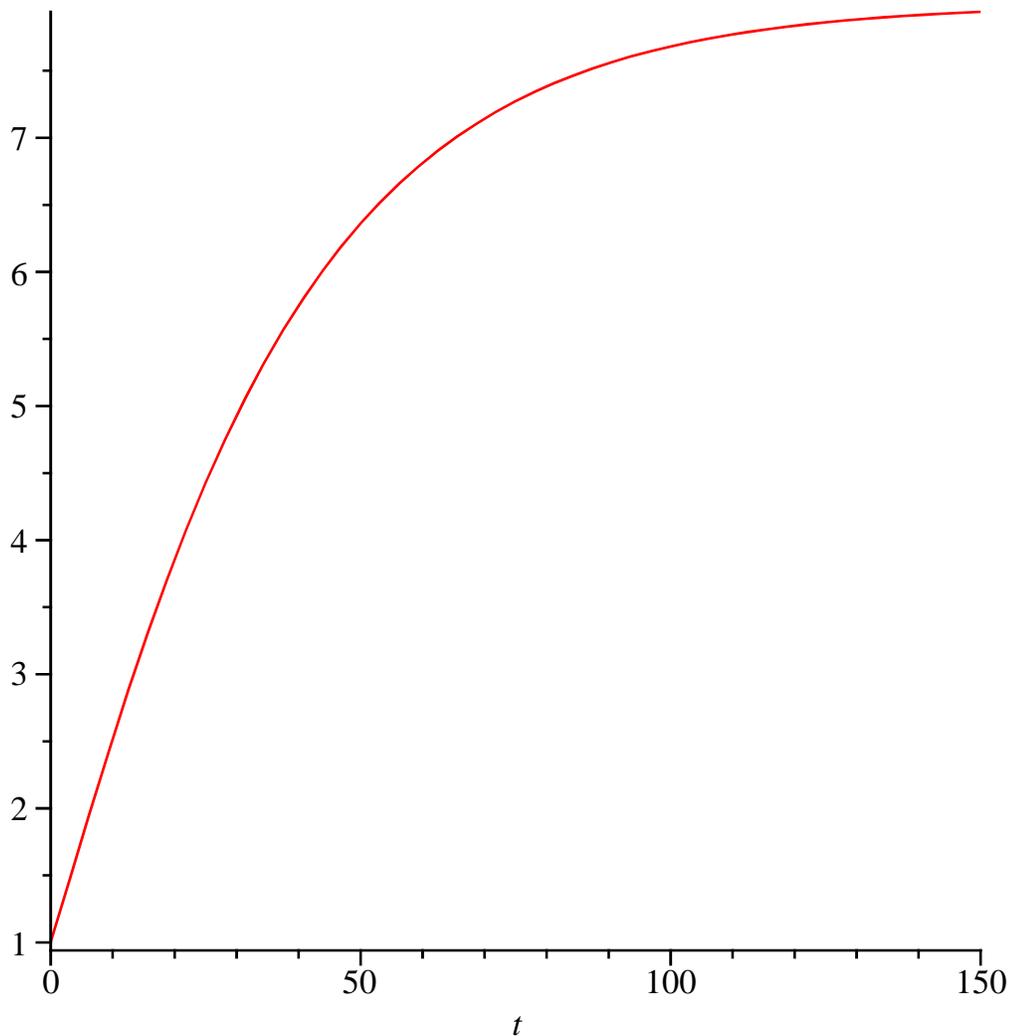
$$\text{soleqd1} := y(t) = e^{\frac{1}{2}t}, y(t) = \frac{1}{\frac{1}{2} + \frac{1}{2} I\sqrt{3}} + \left( \frac{1}{2} + \frac{1}{2} I\sqrt{3} \right) e^t$$

De même, Maple s'en sort plutôt bien avec le modèle de Solow :

```
> restart;
sol:=dsolve({diff(k(t),t)=0.2*k(t)^(1/3)-0.05*k(t),k(0)=
1},k(t));
sol_k:=subs(sol,k(t));
plot(sol_k,t=0..150);
```

$$\text{sol} := k(t) = \left( 4 - 3 e^{-\frac{1}{30}t} \right)^{3/2}$$

$$\text{sol}_k := \left( 4 - 3 e^{-\frac{1}{30}t} \right)^{3/2}$$



Il reste que la commande **dsolve** n'est pas la panacée universelle. Il faut bien comprendre que le logiciel doit être souvent guidé, ce qui suppose une bonne connaissance de la théorie des équations différentielles. Cela dit, dans les cas désespérants et désespérés, on peut recourir à des méthodes de résolution approchée comme on le verra ultérieurement.

┌ ┌ Auparavant, on explore quelques commandes avancées qui facilitent la résolution des EDO.

## ▼ Commandes avancées

┌ Cette sous-section expose quelques "trucs" utiles pour vérifier une solution, documenter un problème et exploiter les possibilités de Maple dans le domaine du "conseil".

### ▼ Vérification de la solution

┌ Il est prudent de vérifier que la solution renvoyée par Maple est correcte. La première méthode est naturellement de remplacer la solution proposée par Maple dans l'équation différentielle de départ au moyen de la commande **eval** et de voir si on obtient bien une équation triviale.

```
> restart;  
eqd:=diff(y(t),t)-2*t*y(t)=(1-2*t)*exp(t);#équation  
différentielle à résoudre  
soleqd:=dsolve(eqd,y(t));#utilisation de dsolve
```

$$eqd := \frac{d}{dt} y(t) - 2 t y(t) = (1 - 2 t) e^t$$

$$soleqd := y(t) = (e^{-t(-1+t)} + \_CI) e^{t^2}$$

```
> eval(eqd,soleqd);#la commande eval remplace dans l'EDO de  
départ la fonction inconnue y(t) par la solution fournie  
par dsolve
```

```
simplify(eval(eqd,soleqd));#on force la simplification
```

$$(1 - 2 t) e^{-t(-1+t)} e^{t^2} = (1 - 2 t) e^t$$
$$- (-1 + 2 t) e^t = - (-1 + 2 t) e^t$$

┌ Le fait d'obtenir une équation triviale signale que la solution proposée est correcte.

┌ Une seconde méthode de vérification consiste à appeler la procédure **odetest(solution Maple, équation différentielle)**. Si la solution convient, la réponse est 0, sinon Maple renvoie l'évaluation du membre de gauche de l'équation différentielle.

```
> odetest(soleqd,eqd);
```

0

┌ La solution est correcte. Ce n'est pas le cas dans l'exemple suivant :

```
> restart;eqd1:=diff(y(t),t)-y(t)=(1-2*t)*exp(t);  
soleqd1:=y(t)=exp(t);#on propose une solution hasardeuse  
odetest(soleqd1,eqd1);
```

$$eqd1 := \frac{d}{dt} y(t) - y(t) = (1 - 2 t) e^t$$

$$soleqd1 := y(t) = e^t$$

$$-e^t + 2 e^t t$$

┌ Le résultat du test est sans appel.

### ▼ Simplification de la rédaction

┌ Il est possible d'écrire des inputs comme on le ferait "à la main", tout comme il est possible d'avoir des réponses respectant l'écriture usuelle.

Pour éviter d'écrire  $y(x)$  quand  $y$  suffit usuellement, on fera appel à la procédure **alias** ( $y=y(x)$ ). En règle générale, la création d'un alias sert à faciliter la rédaction d'un input.

```
> restart; alias(y=y(x)); eqd:=2*diff(y,x)+5*y=cos(x); dsolve
(eqd,y);
```

$$eqd := 2 \left( \frac{\partial}{\partial x} y \right) + 5 y = \cos(x)$$

$$y = \frac{5}{29} \cos(x) + \frac{2}{29} \sin(x) + e^{-\frac{5}{2}x} \_C1$$

On peut même aller plus loin et simplifier également l'écriture de la dérivée première en input. A noter que Maple refuse  $y'$  car l'apostrophe est un symbole réservé; il faut alors passer l'écriture  $\text{'y'}$  à l'aide de la combinaison de touches "Ctrl+Alt+è" suivie d'un espace puis par "y+apostrophe" enfin par la même combinaison "Ctrl+Alt+è+espace".

```
> restart: alias(y=y(x), 'y'=diff(y(x),x)); eqd:=2*'y'+5*y=
cos(x); dsolve(eqd,y);
```

$$eqd := 2 y' + 5 y = \cos(x)$$

$$y = \frac{5}{29} \cos(x) + \frac{2}{29} \sin(x) + e^{-\frac{5}{2}x} \_C1$$

### Recherche d'informations sur les équations différentielles

Si on désire obtenir des renseignements sur la nature de l'équation différentielle étudiée, on peut faire appel à la commande **odeadvisor** du package **DEtools**.

```
> restart;
eqdif1:=sin(x)*diff(y(x),x)-cos(x)*y(x)=0;#exemple 1
with(DEtools):odeadvisor(eqdif1);
```

$$eqdif1 := \sin(x) \left( \frac{d}{dx} y(x) \right) - \cos(x) y(x) = 0$$

[\_separable]

```
eqdif2:=diff(y(x),x)=(a-y(x))(b-y(x));#exemple 2
odeadvisor(eqdif2);
```

$$eqdif2 := \frac{d}{dx} y(x) = a(b - y(x)) - y(x)(b - y(x))$$

[\_quadrature]

La réponse fournie par **odeadvisor** suggère une méthode de recherche qu'on peut exploiter pour extraire la solution. Pour connaître la liste des équations différentielles reconnues par la version du logiciel, on se référera à l'aide suivante :

```
> ?DEtools[odeadvisor]
```

Si on veut suivre pas à pas les étapes de la résolution, on peut commencer la session de travail par la déclaration **infolevel[dsolve]:=N**, où **N** est un entier naturel de préférence égal à 2 ou 3. Par défaut, **N** vaut 0 et Maple ne délivre pas d'information sur les étapes de résolution.

```

> restart;
infolevel[dsolve]:=3;eqd:=y(x)*diff(y(x),x)=sqrt(y(x)^2
-2*y(x)+1);
soleqd:=dsolve(eqd,y(x));

```

$infolevel_{dsolve} := 3$

$$eqd := y(x) \left( \frac{d}{dx} y(x) \right) = \sqrt{y(x)^2 - 2y(x) + 1}$$

Methods for first order ODEs:

```
--- Trying classification methods ---
```

```
trying a quadrature
```

```
trying 1st order linear
```

```
trying Bernoulli
```

```
trying separable
```

```
<- separable successful
```

$$soleqd := x - \frac{(y(x) - 1) (y(x) + \ln(y(x) - 1))}{\sqrt{(y(x) - 1)^2}} + \_CI = 0$$

A l'évidence, cette équation est particulièrement gratinée. Le logiciel a détecté une équations à variables séparables. Prudemment, il indique qu'il est aisé de déterminer  $x$  en fonction de  $y(x)$ , mais la réciproque ne va pas de soi.

Etudions une fois encore l'équation d'accumulation du capital par tête du modèle de Solow. En premier lieu, on prend un cas particulier :

```

> eqsolow:=diff(y(x),x)=0.2*y(x)^(1/3)-0.05*y(x);solsolow:=
dsolve(eqsolow,y(x));

```

$$eqsolow := \frac{d}{dx} y(x) = 0.2 y(x)^{1/3} - 0.05 y(x)$$

Methods for first order ODEs:

```
--- Trying classification methods ---
```

```
trying a quadrature
```

```
trying 1st order linear
```

```
trying Bernoulli
```

```
<- Bernoulli successful
```

$$solsolow := y(x)^{2/3} - 4 - e^{-\frac{1}{30}x} \_CI = 0$$

Maple a reconnu une équation de Bernouilli, dont il donne une solution implicite. Quid du cas général?

```

> eqsolowg:=diff(y(t),t)=s*y(t)^(alpha)-n*y(t);solsolowg:=
dsolve(eqsolowg,y(t));

```

$$eqsolowg := \frac{d}{dt} y(t) = s y(t)^\alpha - n y(t)$$

Methods for first order ODEs:

```
--- Trying classification methods ---
```

```
trying a quadrature
```

```
trying 1st order linear
```

```
trying Bernoulli
<- Bernoulli successful
```

$$solslowg := y(t) = \frac{1}{\left( \frac{s + e^{n(\alpha-1)t} C_1 n}{n} \right)^{\frac{1}{\alpha-1}}}$$

Non seulement la nature de l'EDO est reconnue, mais en plus la solution générale explicite est retournée.

## ▼ Méthode de résolution approchée

Quand **dsolve** ne sait pas résoudre une équation différentielle, elle renvoie un écho ... vide.

```
> restart; dsolve(2*y(x)^2*diff(y(x),x)-y(x)+x=0,y(x));
```

Par **infolevel[dsolve]**, on peut se convaincre que Maple a bien cherché la solution, mais en vain.

```
> restart; infolevel[dsolve]:=3; dsolve(2*y(x)^2*diff(y(x),x)-y(x)+x=0,y(x));
```

```
infoleveldsolve := 3
```

Methods for first order ODEs:

```
--- Trying classification methods ---
```

```
trying a quadrature
```

```
trying 1st order linear
```

```
trying Bernoulli
```

```
trying separable
```

```
trying inverse linear
```

```
trying homogeneous types:
```

```
trying Chini
```

```
differential order: 1; looking for linear symmetries
```

```
trying exact
```

```
Looking for potential symmetries
```

```
trying inverse_Riccati
```

```
trying an equivalence to an Abel ODE
```

```
trying 1st order ODE linearizable_by_differentiation
```

```
--- Trying Lie symmetry methods, 1st order ---
```

```
-> Computing symmetries using: way = 2
```

```
-> Computing symmetries using: way = 3
```

```
-> Computing symmetries using: way = 4
```

```
trying symmetry patterns for 1st order ODEs
```

```
-> trying a symmetry pattern of the form [F(x)*G(y), 0]
```

```
-> trying a symmetry pattern of the form [0, F(x)*G(y)]
```

```
-> trying symmetry patterns of the forms [F(x),G(y)] and [G(y),F(x)]
```

```

-> trying a symmetry pattern of the form [F(x),G(x)]
-> trying a symmetry pattern of the form [F(y),G(y)]
-> trying a symmetry pattern of the form [F(x)+G(y), 0]
-> trying a symmetry pattern of the form [0, F(x)+G(y)]
-> trying a symmetry pattern of the form [F(x),G(x)*y+H(x)]
-> trying a symmetry pattern of conformal type

```

Alors, en ajoutant l'option **numeric** dans la commande **dsolve** et à condition d'ajouter **suffisamment de conditions initiales de sorte qu'aucune constante d'intégration n'intervienne dans les calculs**, on obtient une **fonction-procédure** qui permet d'obtenir des valeurs numériques à partir de la méthode éprouvée dite de Runge-Kutta-Fehlberg au 4<sup>o</sup> et 5<sup>o</sup> ordre (en abrégé *rkf45*) et de représenter graphiquement la solution.

```

> infolevel[dsolve]:=0;#pour éviter l'affichage d'informations
sur la résolution.
soluce:=dsolve({2*y(x)^2*diff(y(x),x)-y(x)+x=0,y(0)=1},y(x),
numeric);

                infoleveldsolve:=0
                soluce:=proc(x_rkf45) ... end proc

```

A partir de cette procédure, on peut demander des évaluations par tâtonnement puisqu'on ne sait rien de la fonction-solution, en particulier de son domaine de définition. Ce dernier se déduit "à l'aveugle" à partir des messages d'erreur renvoyés par Maple.

```

> soluce(5);
soluce(2.5);
Error, (in soluce) cannot evaluate the solution further right
of 2.7492250, probably a singularity
[x=2.5, y(x)=0.897148709247561072]

```

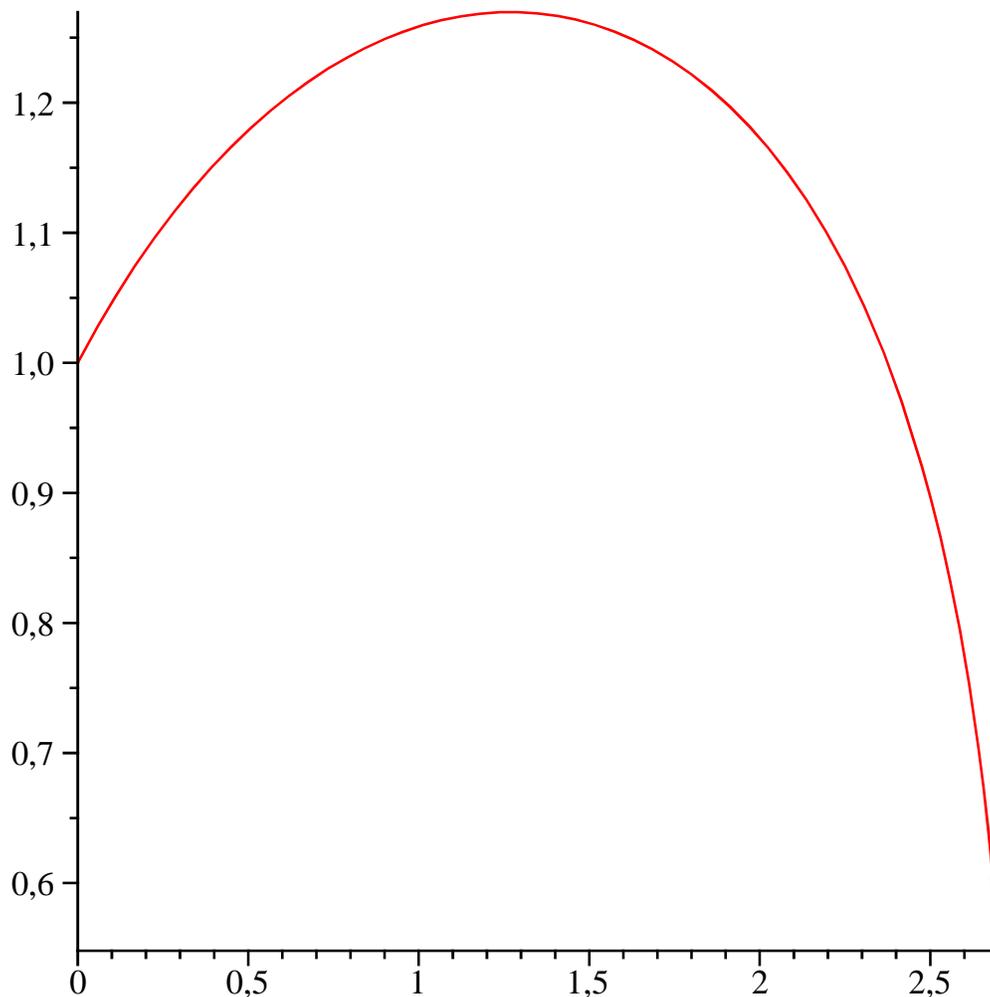
La procédure renvoie une liste de deux égalités. Logiquement, en récupérant les valeurs prises par  $y(x)$  pour différentes valeurs de  $x$ , on peut demander une représentation graphique qui tient compte des indications obtenues par tâtonnements sur le domaine de définition. On crée d'abord une fonction dont l'argument est différent de  $x$ , puis on la trace avec **plot**.

```

> f:=t->subs(soluce(t),y(x));
plot(f,0..2.7);

                f:=t->subs(soluce(t),y(x))

```



## ▼ Méthode graphique de résolution : **DEplot**

Soit une équation différentielle d'ordre 1 écrite sous forme résolue  $y' = F(x, y)$ . L'instruction **DEplot** permet d'en tracer des courbes intégrales correspondant à un ensemble de conditions initiales données. Elle se situe dans le paquetage **DEtools**, qu'il faut préalablement charger.

Sa syntaxe minimale requiert quatre arguments obligatoires :

1° l'équation différentielle étudiée, écrite sous la forme **diff(y(x), x) = F(x, y(x))** ou **diff(y(x), x) - F(x, y(x))** ou encore **D(y)(x) = F(x, y(x))** ;

2° la fonction inconnue **y(x)**

3° L'intervalle de variation de la variable  $x$  écrite sous la forme **x=a..b** ou même **a..b**

4° une liste de conditions initiales, chacune d'elles étant une liste de deux valeurs dont la première est celle de la variable  $x$  et la seconde la valeur prise par la fonction  $y$  en ce point. Comme une liste Maple est placée entre crochets, on a donc : **[[x0=a0, y0=a0], [x1=a1, y1=b1], ..., [xn=an, yn=bn]]** où **n** est un entier éventuellement nul (cas où on demande le tracé d'une seule courbe) et où certains voire tous les **xi** peuvent être égaux entre eux.

Au total, l'instruction minimale a une de ces trois formes :

**DEplot(diff(y(x), x) = F(x, y(x)), [x, y], x=a..b, [[x0=a0, y0=a0], [x1=a1,**

```

y1=b1],..., [xn=an,yn=bn]])
DEplot(diff(y(x),x)-F(x,y(x)), [x,y], x=a..b, [[x0=a0,y0=a0], [x1=a1,
y1=b1], ..., [xn=an,yn=bn]])
DEplot(D(y)(x)=F(x,y(x)), [x,y], x=a..b, [[x0=a0,y0=a0], [x1=a1,
y1=b1], ..., [xn=an,yn=bn]])

```

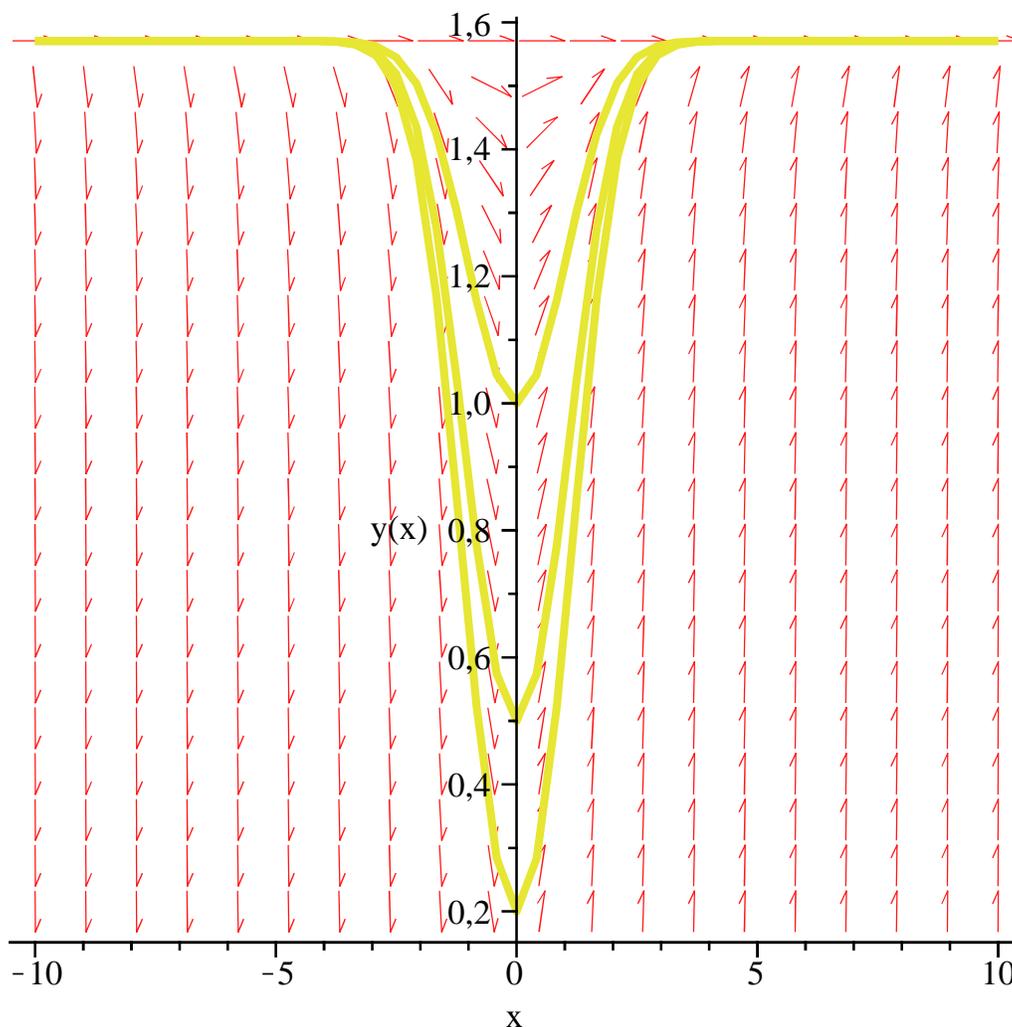
L'exemple suivant concerne l'EDO  $y' = x \cos(y)$ . On suppose qu'on cherche moins sa solution explicite que l'allure de la trajectoire/courbe intégrale.

```

> restart;
with(DEtools):
eqd:=diff(y(x),x)=x*cos(y(x));
DEplot(eqd,y(x),x=-10..10,[[0,1],[0,0.5],[0,0.2]]);

```

$$eqd := \frac{d}{dx} y(x) = x \cos(y(x))$$



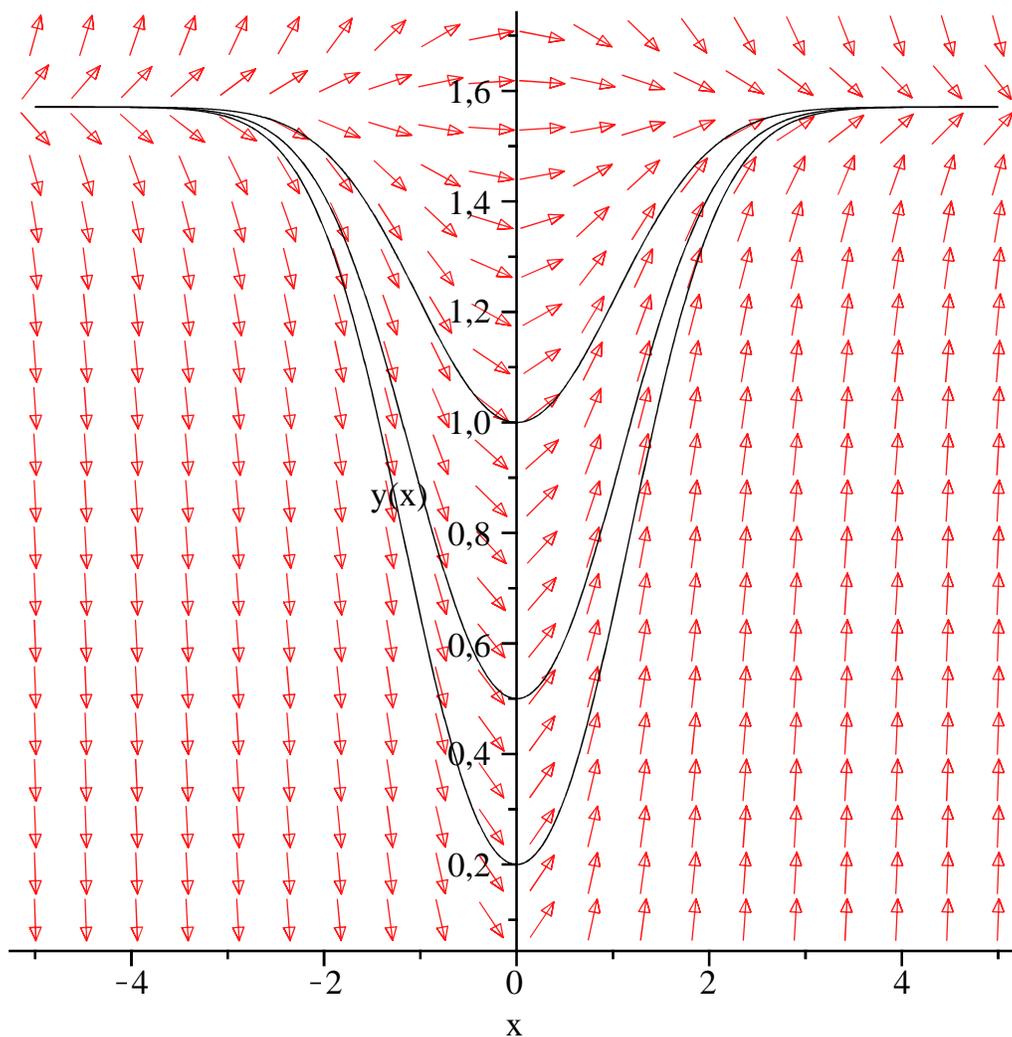
L'ajout d'arguments optionnels est quasiment obligatoire pour obtenir un tracé satisfaisant. Insistons sur les points suivants :

1. Ajustement de l'intervalle de  $y$  par l'option **y=c..d**;
2. Paramétrage de l'épaisseur du trait de la courbe avec l'option **thickness** et de sa couleur avec l'option **linecolor**;

3. Paramétrage des flèches avec l'option **arrows=SMALL, MEDIUM, LARGE, LINE** (absence de pointe), ou **NONE** (pas de flèches);
4. Surtout, amélioration de la précision avec l'option **stepsize=N** où **N** est un réel précisant la manière dont il faut subdiviser l'intervalle des abscisses. Par défaut, la subdivision utilisée par Maple vaut  $\frac{|b-a|}{20}$ . Elle est souvent très insuffisante et il convient de lui attribuer une valeur plus faible.
5. En outre, plusieurs options de présentation, proches de ce qu'on trouve dans **plot**, sont disponibles, par exemple **linecolor**.

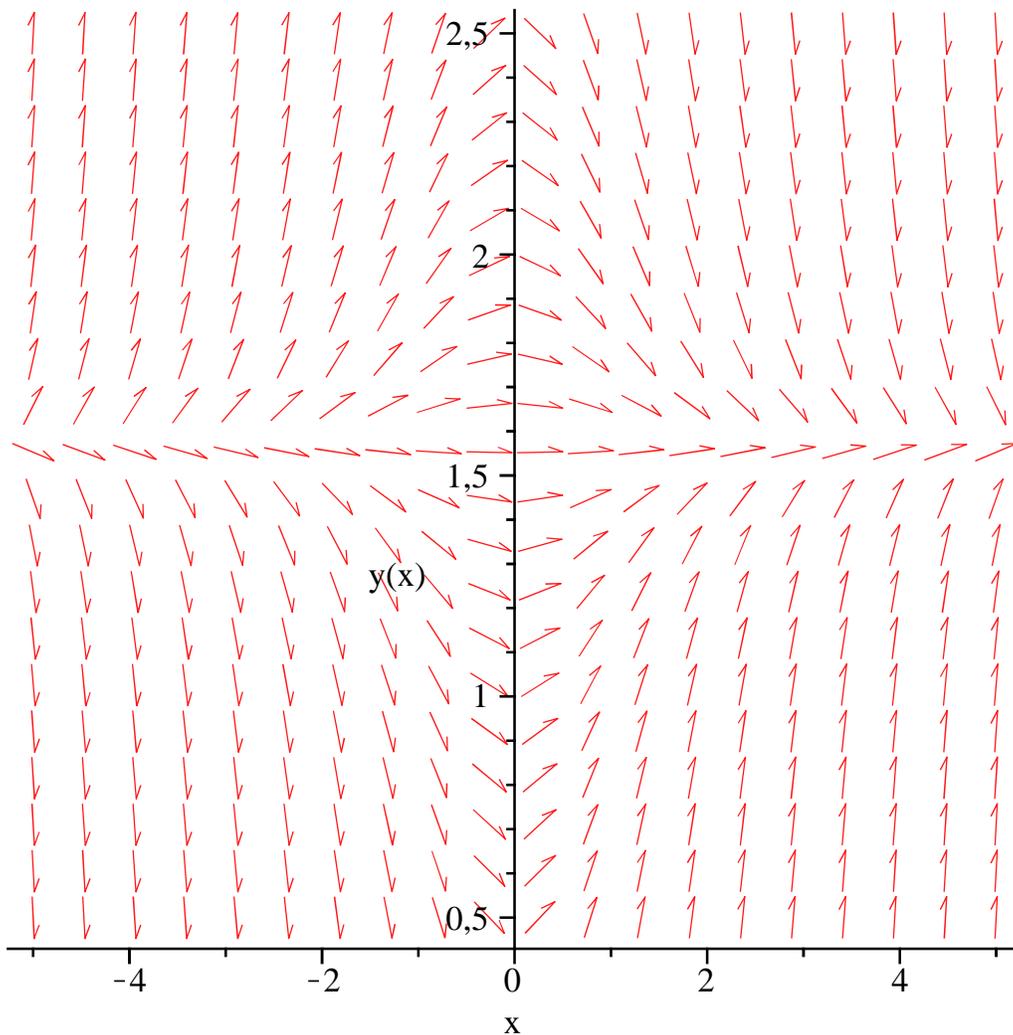
Comparez le tracé précédent et celui-ci, bien plus "intéressant" :

```
> DEplot(eqd,y(x),x=-5..5,[[0,1],[0,0.5],[0,0.2]],y=0.1..1.7,
stepsize=0.01,thickness=0,linecolor=black,arrows=MEDIUM);
```



Si on désire uniquement le champ de vecteurs, il faut utiliser l'instruction **dfieldplot** qui se trouve elle aussi dans le paquetage **DEtools**.

```
> with(DEtools):dfieldplot(eqd,y(x),x=-5..5,y=0.5..2.5);
```

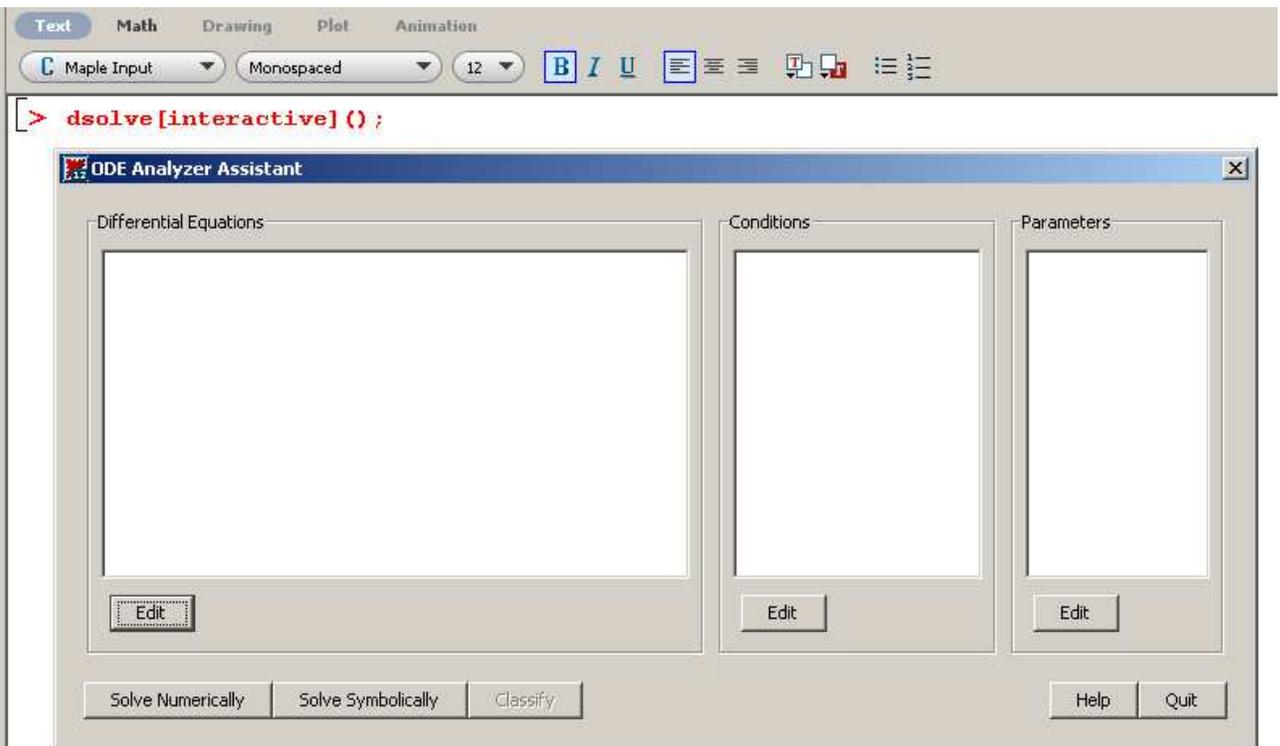


## ▼ Interactivité

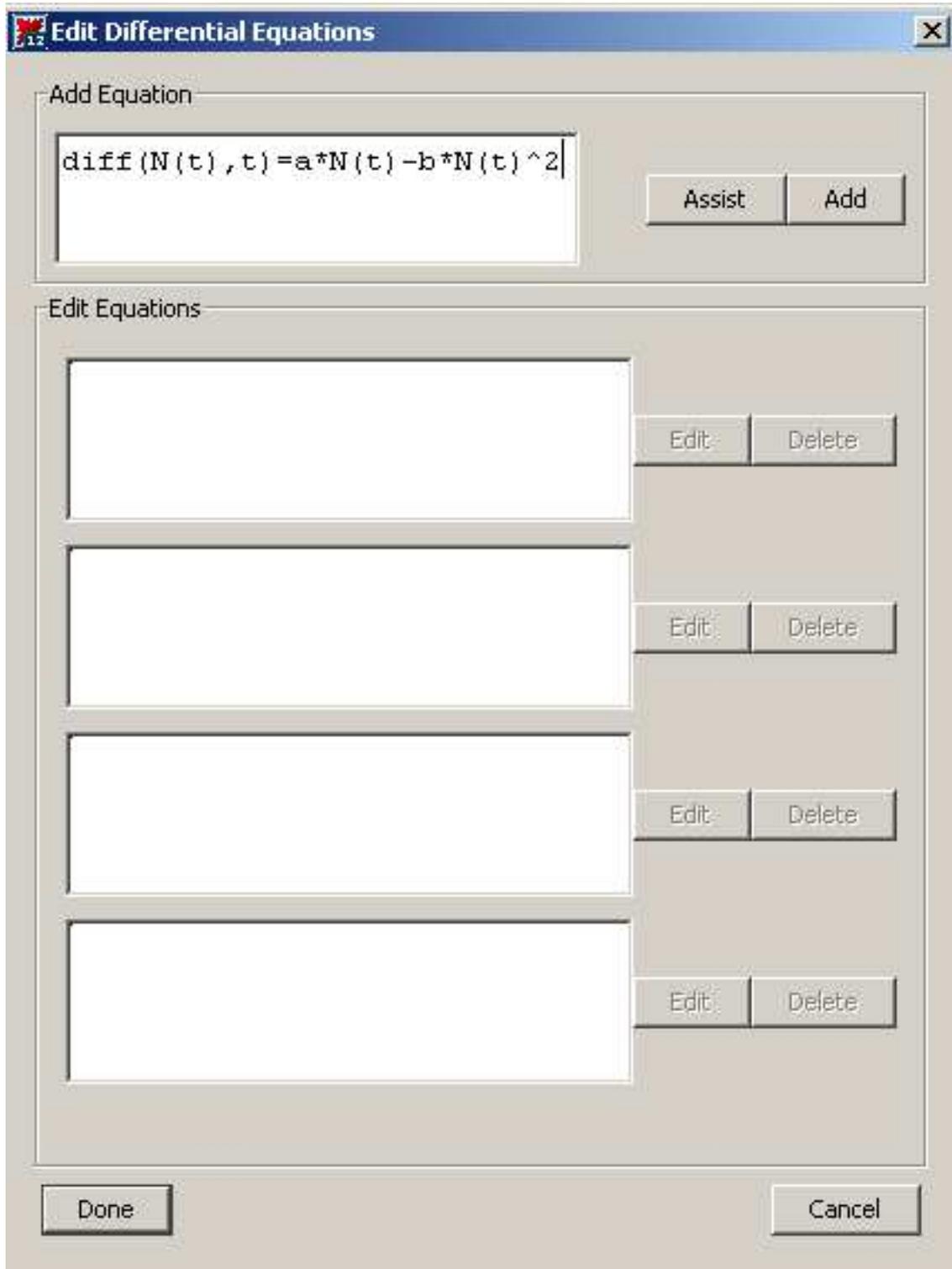
Dans les versions récentes de Maple, il n'est plus obligatoire de programmer la résolution d'une EDO puisqu'on dispose d'un assistant interactif dédié grâce auquel il suffit de remplir un questionnaire pour connaître et visualiser la solution (si elle existe). Cet outil est disponible par Tools → Assistants → ODE Analyzer... ou en frappant la commande suivante :

```
> dsolve[interactive]();
```

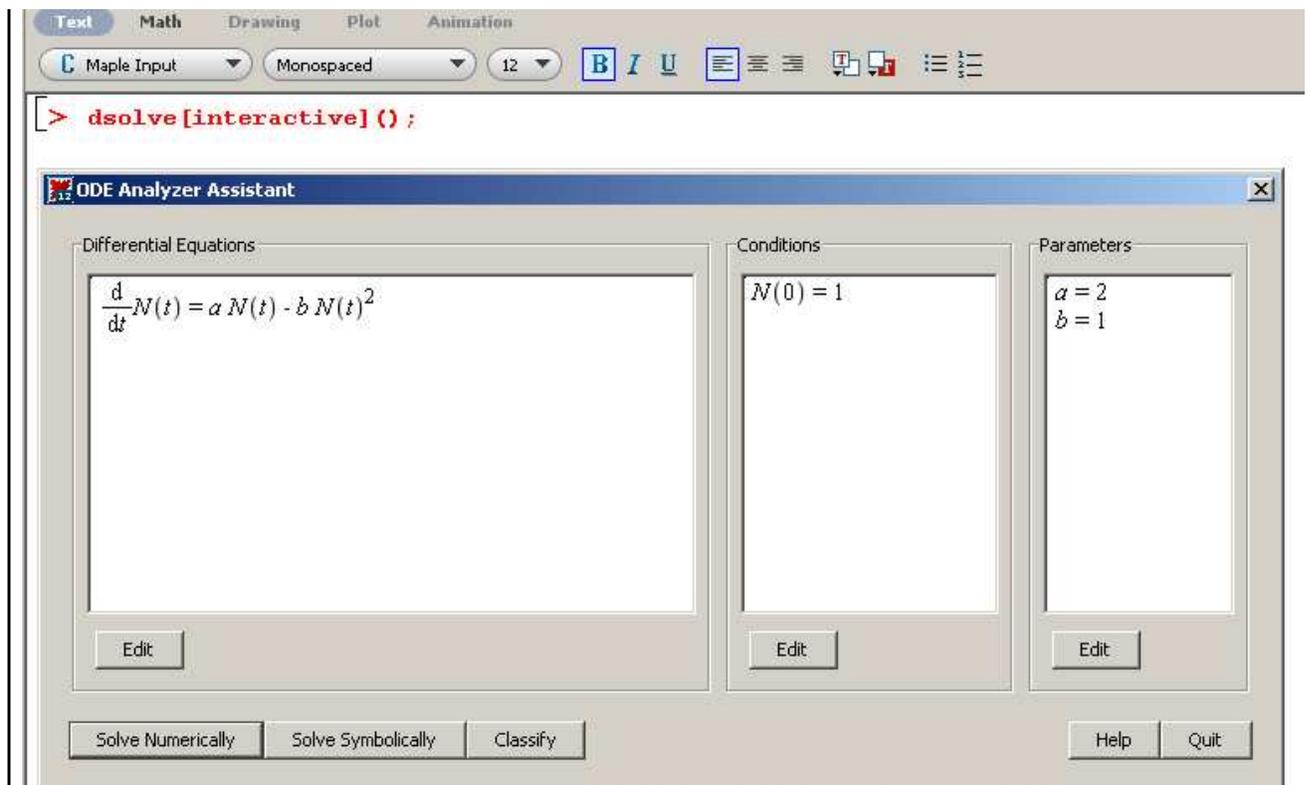
L'assistant se présente comme une fenêtre intitulée ODE Analyzer Assistant :



L'équation différentielle à étudier doit figurer dans le champ Differential Equations. Le bouton Edit fait accéder à une sous-fenêtre intitulée Edit Differential Equations. Le cadre Add Equation permet d'écrire l'équation différentielle considérée (en se faisant éventuellement aider par l'assistant Assist pour écrire correctement les dérivées), soit, dans l'exemple qui suit, l'équation caractéristique d'un modèle auto-freiné :  $N'(t) = a N(t) - b N(t)^2$  avec  $a$  et  $b$  réels strictement positifs et  $N(0) \in \left]0, \frac{a}{b}\right[$ . Un clic sur Add transfère l'EDO dans le premier cadre du champ Edit Equations. Si le résultat est satisfaisant, un clic sur Done renvoie à la fenêtre d'accueil où l'équation est maintenant éditée.



Les champs Conditions et Parameters se remplissent de la même manière. Une fois ce premier questionnaire rempli, on obtiendra par exemple :



L'étape de résolution peut alors commencer. Le bouton Help informe que l'utilisateur peut demander une résolution numérique (approchée) via Solve Numerically, ou une résolution symbolique (exacte) via Solve Symbolically, ou encore des informations sur le type de l'EDO via Classify.

En cliquant sur le bouton Solve Numerically, on accède à la fiche-fenêtre correspondante. Après avoir documenté les champs, on aura par exemple :

Text Math Drawing Plot Animation

Maple Input Monospaced 12 B I U

> dsolve[interactive]();

ODE Analyzer Assistant

Solve Numerically

Parameters

- Runge-Kutta-Fehlberg 4-5th order
- Dverk 7-8th order
- Gear single step extrapolation
- Rosenbrock stiff 3-4th order
- Livermore stiff
- Boundary Value Problem solver
- 
- Range of t:  to
- Taylor series
- Modified Extended BDF Implicit
- Fixed step methods
- 
- Absolute:  default
- Relative:  default

Output

Show function values at t =

N = 1.46211731669484380

Show Maple commands

```
soll := dsolve([diff(N(t),t) =
2*N(t)-N(t)^2, N(0) = 1], numeric);
soll(.5);
plots[odeplot](soll, 0..10, color = red);
```

On Quit, Return

De même, en cliquant sur le bouton Solve Symbolically, on accède à la fiche-fenêtre éponyme qui permet d'obtenir :

The screenshot shows the Maple ODE Analyzer Assistant interface. The main window displays the differential equation  $\frac{d}{dt}N(t) = 2N(t) - N(t)^2$ . The 'Solve Symbolically' dialog is open, showing the 'Default' method selected. The 'Output' section displays the solution  $N(t) = \frac{2}{1 + e^{-2t}}$  and a plot of this function. The plot shows a red curve starting at (0, 1) and asymptotically approaching 2 as t increases. The Maple commands for solving and plotting are shown in a text area at the bottom of the dialog.

On sort des ces fiches en cliquant sur Quit. Suivant l'option retenue dans le champ On Quit, Return, la solution ou la courbe intégrale ou les commandes Maple s'affichent dans la feuille de travail.

## Exercices

### Exercice M1

Déterminer la solution générale des EDO linéaires du premier ordre suivantes.

- 1)  $y'(t) + 2y(t) = 4t^2$
- 2)  $y'(t) + y(t) = 2e^t$
- 3)  $y'(t) - 2y(t) = 2t^3 + t$
- 4)  $y'(t) - 3y(t) = 2e^{3t}$
- 5)  $y'(t) - y(t) = t + e^t$
- 6)  $y'(t) - 2y(t) = 4\cos(2t)$

$$7) y'(t) + y(t) = 2t e^{-t}$$

### Exercice M2

Déterminer la solution générale des EDO linéaires du premier ordre suivantes.

$$1) y'(t) - 2t y(t) = (1 - 2t) e^t$$

$$2) y'(t) - 2t y(t) = \sin(t) e^{2t}$$

$$3) y'(t) - \cos(t) y(t) = \cos(t)$$

$$4) y'(t) - \cos(t) y(t) = \sin(2t)$$

$$5) y'(t) \sin(t) - \cos(t) y(t) = 3t^2 \sin(t)^2$$

$$6) (1 - t^2) y'(t) - 2t y(t) = t^2$$

$$7) (1 + t^2) y'(t) + t y(t) - 2t = 0$$

$$8) t(-1 + t)y'(t) - (2t - 1)y(t) + t^2 = 0$$

### Exercice M3

Résoudre les problèmes de Cauchy suivants. Donner à chaque fois une représentation de la trajectoire-solution.

$$1) y'(t) + 2y(t) = 4t^2 \text{ avec } y(0) = 3.$$

$$2) y'(t) + y(t) = 2e^t \text{ avec } y(0) = 0.$$

$$3) y'(t) - 2y(t) = 2t^3 + t \text{ avec } y(0) = 1.$$

$$4) y'(t) - 3y(t) = 2e^{3t} \text{ avec } y(0) = 4e^3.$$

$$5) y'(t) - y(t) = t + e^t \text{ avec } y(0) = -1.$$

$$6) y'(t) - 2y(t) = 4 \cos(2t) \text{ avec } y(0) = \frac{17}{20}.$$

$$7) y'(t) + y(t) = 2t e^{-t} \text{ avec } y(0) = 1.$$

### Exercice M4

Résoudre les problèmes de Cauchy suivants. Donner à chaque fois une représentation de la trajectoire-solution.

$$1) y'(t) - 2t y(t) = (1 - 2t) e^t \text{ avec } y(0) = 5.$$

$$2) y'(t) - 2t y(t) = \sin(t) e^{2t} \text{ avec } y(0) = 1.$$

$$3) y'(t) - \cos(t) y(t) = \cos(t) \text{ avec } y(0) = 0.$$

$$4) y'(t) - \cos(t) y(t) = \sin(2t) \text{ avec } y(0) = -1.$$

$$5) y'(t) \sin(t) - \cos(t) y(t) = 3t^2 \sin(t)^2 \text{ avec } y(1) = 0.$$

$$6) (1 - t^2) y'(t) - 2t y(t) = t^2 \text{ avec } y(0) = 1.$$

$$7) (1 + t^2) y'(t) + t y(t) - 2t = 0 \text{ avec } y(1) = 3.$$

$$8) t(-1 + t)y'(t) - (2t - 1)y(t) + t^2 = 0 \text{ avec } y(2) = 1.$$

### Exercice E1 (dynamique du prix de marché)

On considère un marché très générique sur lequel l'offre agrégée et la demande agrégée dépendent respectivement positivement et négativement du prix courant. Dans la version linéarisée du modèle, on a :  $Qs(t) = aP(t) - b$  et  $Qd(t) = -cP(t) + d$ , tous les paramètres étant positifs. La demande excédentaire à la date  $t$  est la différence entre l'offre et la demande courantes :  $Ed(t) = Qd(t) - Qs(t)$ . L'ajustement du prix obéit à la règle suivante : un excès positif de demande accroît le prix; un excès de demande négatif - autrement dit un excès d'offre -

fait baisser le prix; l'égalité de l'offre et de la demande laisse le prix au même niveau. La variation instantanée du prix s'interprétant comme la dérivée première de  $P(t)$  par rapport à  $t$ , la traduction formelle la plus simple de ce principe est linéaire :  $\frac{d}{dt} P(t) = m Ed(t)$ , le paramètre  $m > 0$  mesurant l'intensité de la réaction du mécanisme d'ajustement du prix au déséquilibre courant.

1) Donner l'expression de la trajectoire-solution du prix de marché à partir de la condition initiale  $P(0) = P_0$ .

2) En microéconomie, la question est moins de savoir calculer la trajectoire suivie par le prix que de savoir s'il existe un prix d'équilibre - assurant par définition l'égalité des volumes offerts et demandés sur le marché - et si cet équilibre est stable - assurant qu'un écart du prix courant au prix d'équilibre se résorbe par le seul jeu des mécanismes de marché. Déterminer le prix d'équilibre puis étudier sa nature. Tracer quelques trajectoires confirmant la convergence de toute trajectoire-solution vers le prix d'équilibre.

### Exercice E2 (dynamique du prix de marché)

Reprendre les questions de l'exercice précédent en supposant à présent que la demande dépend négativement du prix courant mais aussi des anticipations parfaites de variations du prix.

Autrement dit, on pose :  $Qd(t) = -c P(t) + d + e \frac{d}{dt} P(t)$  avec  $e > 0$ .

### Exercice E3 (démographie)

Le modèle démographique de Verhulst repose sur l'idée simple qu'en l'absence de mouvements migratoires une population varie selon son taux d'accroissement naturel, défini comme la différence entre le taux de natalité et le taux de mortalité. Le taux de natalité  $n(t)$ , rapport entre le nombre de naissances et la population totale  $P(t)$ , évolue suivant la règle  $n(t) = a - b P(t)$  avec  $a$  et  $b$  positifs : plus la taille de la population est élevée, plus le taux de natalité est faible. Le taux de mortalité  $m(t)$ , rapport entre le nombre de décès et la population totale  $P(t)$ , évolue suivant la règle  $m(t) = c + d P(t)$  avec  $c$  et  $d$  positifs : plus la taille de la population est élevée, plus le taux de mortalité est élevé.

1) Montrer que l'évolution de la population obéit à une loi de Bernoulli. Donner alors l'expression de la trajectoire pour la condition initiale  $P(0) = P_0$ .

2) Montrer qu'il existe deux états stationnaires pour la dynamique démographique.

3) Montrer que la convergence vers un de ces deux états dépend des paramètres  $a$  et  $c$ .

### Exercice E4 (problème de Constantin)

Constantin Chilarescu est un économiste-mathématicien roumain spécialisé dans les modèles de croissance endogène. Il a souvent recours à Maple pour confirmer ses résultats théoriques, voire pour l'aider à résoudre des problèmes complexes. Le but de cet exercice est d'étudier une extension qu'il a donnée au modèle de Lucas-Uzawa et qui conduit à la forme réduite

$$\frac{d}{dt} z(t) = \frac{p z(t)}{b - a} - \frac{g z(t)^{2-b}}{(1-a)(b-a)} + \frac{d(1-a) z(t)^{1-a}}{b-a} \text{ où } z \text{ est le capital par unité de}$$

travail efficace,  $a$  est l'élasticité du capital ( $0 < a < 1$ ),  $b$  est l'élasticité du travail ( $0 < b < 1$  et  $a < b$ ),  $d$  est le taux de dépréciation du capital ( $0 < d < 1$ ),  $p$  est le taux de préférence pour le présent ( $p > 0$ ) et  $g$  est le taux de croissance du progrès technique ( $g > 0$ ).

1) Mener une étude qualitative de la dynamique d'accumulation du capital en étudiant les propriétés de la fonction :

$$f(z) = \frac{p z}{b-a} - \frac{g z^{2-b}}{(1-a)(b-a)} + \frac{d(1-a) z^{1-a}}{b-a}$$

2) On donne aux paramètres les valeurs suivantes :

$a = 0.3$ ;  $b = 0.35$ ;  $d = 0.075$ ;  $g = 1.05$ ;  $p = 0.05$ . Mener une étude qualitative graphique de l'équation différentielle à l'aide de **dsolve** et de **DEplot**.